Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Safe Multi-Agent Navigation via Constrained HJB-Informed Learning

Fenglan Wang, Xinguo Shu, Lei He, and Lin Zhao*
Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583
Email: wfenglan@nus.edu.sg (F. Wang), e1352650@u.nus.edu (X. Shu),
lei.he@nus.edu.sg (L. He), zhaolin@nus.edu.sg (L. Zhao)

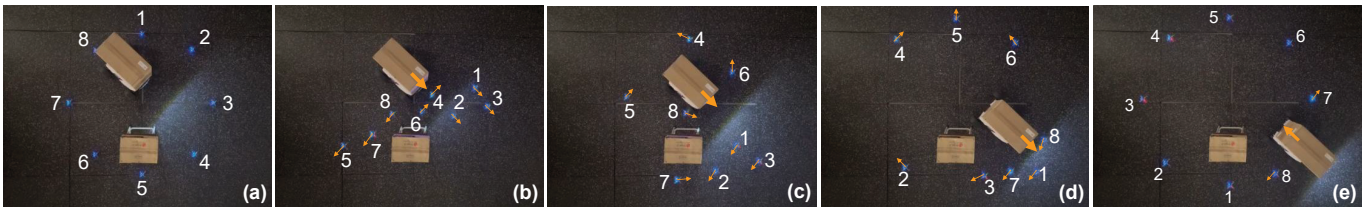

Fig. 1: **Antipodal position swapping of eight Crazyflie miniature drones using the proposed HJB-GNN navigation policy under limited sensing range in an unseen environment containing static and dynamic obstacles.** HJB-GNN is a Hamilton-Jacobi-Bellman (HJB) equation-based learning framework for distributed infinite-horizon safe optimal navigation using graph neural networks (GNN). It enables a principled and adaptive trade-off between goal reaching and collision-avoidance through graph-dependent Lagrange multipliers derived from Karush-Kuhn-Tucker (KKT) optimality conditions. (a) Starting positions. (b)-(e) Temporal snapshots illustrating the evolution of the drone and obstacle trajectories. Orange arrows indicate the velocity directions of the drones and a large, randomly moving dynamic obstacle.

***Abstract*—Multi-agent navigation in unknown and cluttered environments has broad applications, yet remains fundamentally challenging. In particular, dense agent-agent and agent-obstacle reactive interactions can exacerbate the inherent competition between collision-avoidance constraints and goal-reaching objectives. Most existing approaches mitigate this by applying per-step safety filtering on top of a predefined goal-reaching controller or by designing heuristic loss functions that penalizes safety constraints violation gradient. While effective in sparse environments, these methods still suffer from overly-conservative behaviors when interactions become dense. To overcome these limitations, we propose HJB-GNN, a Hamilton-Jacobi-Bellman (HJB)-based learning framework that jointly learns a graph neural network (GNN)-parameterized control barrier function for explicit safety enforcement, a distributed GNN-based navigation policy, and a value function that induces goal-reaching behavior. By exploiting the analytical solution of the constrained HJB equation, the proposed method derives graph-dependent Lagrange multipliers that adaptively balance collision-avoidance and goal-reaching across diverse multi-agent navigation scenarios. Moreover, HJB-GNN supports centralized training with distributed deployment. Extensive simulations and real-world experiments with Crazyflie drone swarms demonstrate its superior safety and goal-reaching performance, as well as strong scalability and generalizability to large-scale teams operating in previously unseen, dense environments.** [1]

I. Introduction

Multi-agent navigation is a core capability underpinning a wide range of real-world robotic systems, including autonomous mobile robots in automated warehouses [26, 31, 16], drone swarms for search-and-rescue operations [35, 24, 3, 19], and unmanned surface vehicles for coordinated patrol missions [10]. Due to limited onboard computation, sensing range, and communication bandwidth, practical deployments typically rely on distributed navigation controllers that use only local neighborhood information [27, 28, 11, 8, 25, 3]. However, in dense, cluttered, and unknown environments, frequent reactive interactions among agents and obstacles substantially intensify the conflict between collision-avoidance constraints and goal-reaching objectives, posing significant challenges for reliable and efficient navigation.

Recent works have attempted to mitigate the conflict between collision-avoidance and goal-reaching objectives through safety filtering. In particular, several approaches [27, 18, 22, 28, 11, 16, 25] employed quadratic programming (QP)-based safety filters, where a predefined nominal goal-reaching controller was often projected onto control barrier function (CBF)-based collision avoidance constraints to ensure safety. However, these works are limited in *known* environments where manually specifying CBF is tractable. For multi-agent systems (MASs) navigating in *unknown* environments under limited sensing range, a line of recent work has focused on learning CBFs directly from data [35, 9]. Notably, [35] proposed graph neural networks (GNNs)-parameterized CBFs (graph CBF or GCBF) for MAS navigation, which naturally adapts to dynamically changing interaction graphs. A distributed GNN-based navigation policy was then learned by imitating the corresponding safety-filtered controller. While it (hereafter referred to as QP-GCBF) demonstrates promis-

*Corresponding author
[1]Project website: https://github.com/hublan24/HJB-GNN.

ing empirical performance, its effectiveness is fundamentally constrained by its reliance on a predefined, fixed nominal controller. To alleviate this limitation, [9] incorporated control Lyapunov function (CLF) constraints into the QP-based framework, thereby avoiding explicit specification of a nominal controller. Nevertheless, both [35] and [9] provided only one-step-ahead safety guarantees. The induced control policies are inherently myopic and can drive agents excessively close to obstacles or other agents, often leading to overly-conservative behavior, deadlocks, or even collisions in dense multi-agent scenarios.

Another line of more recent works have further proposed MAS navigation solutions in a constrained infinite-horizon optimal control formulation [33, 34]. In particular, [33] proposed learning discrete-time graph CBF-constrained navigation policies using proximal policy optimization [21] in a model-free constrained reinforcement learning framework. They mitigate the conflicts between safety and goal-reaching (task) in the loss design through an approximate gradient projection mechanism, where task gradients are projected to be orthogonal to the safety constraints gradients. However, their safety-performance trade-off is still regulated by heuristic scheduling of penalty weights, which tends to induce overly-conservative behaviors in dense and cluttered environments due to complex interaction constraints. Furthermore, to enable stable training, [34] developed a distributed learning algorithm based on an epigraph reformulation, in which an auxiliary variable is minimized as an upper bound on the task cost. However, this reformulation achieves collision-avoidance and goal-reaching coordination indirectly through the evolution of the auxiliary variable, which can limit the agent's responsiveness to rapidly changing interactions in dense environments and easily result in deadlocks or collisions.

To tackle these limitations, we propose a novel infinite-horizon CBF-constrained optimal control formulation for safe multi-agent navigation in unknown environments with limited sensing radii. We formally analyze the resulting constrained optimal control problem through the lens of Hamilton-Jacobi-Bellman (HJB) equations and Lagrange duality, and develop a physics-informed learning framework that leverages GNNs to approximate the solution. Unlike existing approaches, which do not explicitly optimize the trade-off between collision-avoidance and goal-reaching, our method exploits the analytical structure of constrained HJB formulation to enable principled, state- and graph-dependent coordination between these competing objectives. As a result, the learned distributed policy effectively handles dense agent-agent and agent-obstacle interactions, substantially reducing deadlocks and collisions in practice (see Fig. 1 for a challenging scenario). More specifically, our main contributions are as follows:

(i) We establish an infinite-horizon safe optimal control framework for MASs using graph CBF. The proposed framework introduces an adaptive trade-off mechanism for safety and goal-reaching performance via graph-dependent Lagrange multipliers. Such graph-dependent multipliers, derived from Karush-Kuhn-Tucker (KKT) optimality conditions, effectively coordinate collision-avoidance and goal-reaching control to adapt to sensing-based changing interaction graph, thereby reducing the conservative behaviors induced by manually tuned static parameter, heuristic gradient projection, or auxiliary variable-based trade-off designs [5, 32, 29, 33, 34].

(ii) We develop a novel HJB-GNN learning algorithm that enables end-to-end joint synthesis of a value function, a graph CBF, and a distributed policy using GNNs, guided by the analytical safe optimal solution structure of constrained HJB equations. The proposed HJB-GNN algorithm eliminates the reliance on short-horizon QP solvers and predefined nominal controllers commonly used in existing safety control methods [35, 25, 9, 6].

(iii) We adopt a centralized training and a distributed deployment paradigm for the HJB-GNN algorithm. Policy trained on systems with only eight agents generalize to scenarios involving hundreds and even thousands of agents, and scale effectively to previously unseen, dense environments. Extensive simulations demonstrate significantly improved safety, scalability, and generalizability compared with the state-of-the-art QP-GCBF method [35]. We validate the applicability and robustness of the HJB-GNN through hardware experiments using Crazyflie drone swarms navigating in complex obstacle environments. [2]

The remainder of the paper is organized as follows. Section II reviews preliminaries. Section III formulates the constrained HJB problem with a graph CBF and analyzes its solution. Section IV presents the HJB-GNN learning framework. Simulations and hardware experiments are given in Sections V. Section VI concludes the paper.

Notations: $\nabla_x W$ represents the derivative of the function W with respect to x and is interpreted as a row vector, and $\nabla_x^T W$ represents the transpose of $\nabla_x W$. A continuous function $\alpha : [-b, a) \to \mathbb{R}$ with $a, b > 0$, is of extended class $\mathcal{K}$ ($\alpha \in \mathcal{K}_e$) if α satisfies $\alpha(0) = 0$ and is strictly increasing. We denote $(a, b) = [a^T, b^T]^T$ for column vectors a and b.

II. Preliminaries

A. System Dynamics

We aim to study the problem of distributed safe multi-agent navigation, where each agent $i \in V_a := \{1, 2, ..., N\}$, is required to reach its goal position while avoiding collisions with other agents and static obstacles. The agents are governed by the following dynamics:

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i) + \mathbf{g}(\mathbf{x}_i)\mathbf{u}_i, \ i = 1, 2, ..., N, \tag{1}$$

where for the i-th agent, $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^n$ and $\mathbf{u}_i \in \mathcal{U} \subset \mathbb{R}^m$ denote the state and control input, respectively, $\mathcal{X}$ and $\mathcal{U}$ are two compact sets, and $\mathbf{f} : \mathcal{X} \to \mathbb{R}^n$, $\mathbf{g} : \mathcal{X} \to \mathbb{R}^{n \times m}$ are two locally Lipschitz continuous functions. Denote $\bar{\mathbf{x}} := (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) \in \mathcal{X}^N$ and $\bar{\mathbf{u}} := (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N)^T \in \mathcal{U}^N$ as the state vector and the control input vector of the MAS (1), respectively.

[2]Simulation and experiment videos can be found at: https://nus-core.github.io/assets/standalone/HJB-GNN/index.html.

Each agent has a *limited* sensing radius $R_a \in \mathbb{R}_{>0}$ and thus communicates with other agents and observes the obstacle environment only within its sensing range. The interactions among agents and obstacles are represented by a direct graph $G = (V, E)$, where $V = V_a \cup V_o$ is the set of nodes which contain the set of agent nodes V_a and the set of obstacle nodes V_o, and $E \subset \{(i,j) | i \in V_a, j \in V\}$ denotes the set of edges. An edge $(i,j) \in E$ implies that the agent i communicates with other agent j or observes an obstacle j. The topology graph G and its edge set E are allowed to vary arbitrarily, without requiring the fixed number of neighbors assumed in [36, 20].

B. Safety of Multi-Agent Systems

Denote a position space by $\mathbb{P} \subset \mathbb{R}^{n_d}$, where $n_d \in \{2, 3\}$ represents the spatial dimensions of environments. In the MAS (1), assume that the first n_d entries of the state vector $\mathbf{x}_i$ correspond to the position $\mathbf{p}_i \in \mathbb{P}$ of agent $i \in V_a$. We denote the vector of relative positions information observed by each agent $i \in V_a$ from all neighboring obstacles within its sensing radius as $\mathbf{O}_i := (\mathbf{O}_{i1}, \mathbf{O}_{i2}, ..., \mathbf{O}_{in_{oi}})$ with $n_{oi} \in \mathbb{N}$. To ensure collision-avoidance among agent-agent and agent-obstacle, denote the overall safe set $\mathcal{S} \subset \mathcal{X}^N$ in the MAS (1) as:

$$\mathcal{S} := \cap_{i=1}^N \mathcal{S}_i, \tag{2a}$$

$$\mathcal{S}_i := \Big\{ \bar{\mathbf{x}} \in \mathcal{X}^N \Big| \big\|\mathbf{O}_{ij}\big\| > r, \forall j = 1, 2, ..., n_{oi}, \min_{i,j \in V_a, i \neq j} \big\|\mathbf{p}_i - \mathbf{p}_j\big\| > 2r \Big\}, \tag{2b}$$

where r represents the radius of a minimal circle enclosing each agent's physical body with $2r < R_a$. The MAS (1) is safe with respect to the set $\mathcal{S}$ if for any initial state $\bar{\mathbf{x}}(t_0) \in \mathcal{S}$, there exists a controller $\bar{\mathbf{u}}$, such that $\bar{\mathbf{x}}(t) \in \mathcal{S}$, $\forall t \geq t_0$.

We employ a graph CBF to deal with safety constraints for the MAS (1) with arbitrarily varying neighborhoods inspired by [35]. For each agent i, let $\mathcal{N}_i$ denote the neighbor set of agent i, that is,

$$\mathcal{N}_i := \big\{ j \in V \big| \|\mathbf{p}_i - \mathbf{p}_j\| \leq R_a, \|\mathbf{O}_{ij}\| \leq R_a \big\}. \tag{3}$$

An augmented neighborhood set $\hat{\mathcal{N}}_i$ consists of M *closest* agents and obstacles if $|\mathcal{N}_i| > M$, and otherwise set $\hat{\mathcal{N}}_i = \mathcal{N}_i$. The corresponding augmented neighborhood state $\bar{\boldsymbol{x}}_{\hat{\mathcal{N}}_i} \in \mathcal{X}^M$ is used as the input to the graph CBF by concatenating the states of agent i and its neighbors, where constant-padding is applied to maintain a fixed dimension when the size of $\hat{\mathcal{N}}_i$ is less than M. $\check{\mathcal{N}}_i \subseteq \hat{\mathcal{N}}_i$ denotes the set of neighbors strictly within R_a. $\hat{\mathcal{N}}_i^a$ and $\check{\mathcal{N}}_i^a$ denote the subsets of agents within $\hat{\mathcal{N}}_i$ and $\check{\mathcal{N}}_i$, respectively. Then, the C^1 graph CBF $h : \mathcal{X}^M \to \mathbb{R}$ is designed such that for all $\bar{\mathbf{x}} \in \mathcal{X}^N$ with $N \geq M$,

$$\begin{aligned} \dot{h}(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) &= \sum_{l \in \hat{\mathcal{N}}_i^a} \nabla_{\mathbf{x}_l} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) (\mathbf{f}(\mathbf{x}_l) + \mathbf{g}(\mathbf{x}_l)\mathbf{u}_l) \\ &\geq -\alpha(h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})), \end{aligned} \tag{4}$$

where $\alpha \in \mathcal{K}_e$, $\mathbf{u}_l = \pi_l(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_l})$ with $\pi_i : \mathcal{X}^M \to \mathcal{U}$, and two conditions are satisfied: $h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) = h(\bar{\mathbf{x}}_{\check{\mathcal{N}}_i})$ and $\nabla_{\mathbf{x}_l} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) = 0$, $\forall l \in \hat{\mathcal{N}}_i^a \setminus \check{\mathcal{N}}_i^a$.

Denote the zero-superlevel set of h by $\mathcal{C} := \{\tilde{\mathbf{x}} \in \mathcal{X}^M \,|\, h(\tilde{\mathbf{x}}) \geq 0\}$. Let $\mathcal{H}_{N,i}$ be the set of states $\bar{\mathbf{x}}$ of MAS where the neighborhood state $\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}$ of agent i is in set $\mathcal{C}$, that is, $\mathcal{H}_{N,i} := \{\bar{\mathbf{x}} \in \mathcal{X}^N \,|\, \bar{\mathbf{x}}_{\hat{\mathcal{N}}_i} \in \mathcal{C}\}$. Denote their intersection as $\mathcal{H}_N := \bigcap_{i=1}^N \mathcal{H}_{N,i}$. Clearly, $\mathcal{H}_{N,i} \subseteq \mathcal{S}_i$ for all $i \in V_a$ implies $\mathcal{H}_N \subseteq \mathcal{S}$. Without loss of generality, assume that there exists a small constant $\varrho \in \mathbb{R}_{>0}$, such that $\|\nabla_{\mathbf{x}_i} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) \mathbf{g}(\mathbf{x}_i)\| \geq \varrho$ for all $\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i} \in \mathcal{X}^M$. It is a regularity condition which makes that the CBF condition (4) is effective on the system control. It can be easily satisfied by a generic GNN.

III. Safe Multi-Agent Navigation via Infinite-Horizon Constrained Optimization

In this section, we formulate an infinite-horizon safe optimal control framework based on a graph CBF. Our framework derives a principled adaptive trade-off mechanism that balances safety and goal-reaching performance via graph-dependent Lagrange multipliers. Crucially, this formulation characterizes the analytical structure of safe optimal controller, providing a rigorous theoretical foundation for the HJB-GNN learning algorithm developed in Section IV.

A. Safe Optimal Control Formulation

To address the safe multi-agent navigation problem without requiring predefined task controllers used in [35, 32, 14], we unify safety and goal-reaching objectives within a single infinite-horizon constrained optimal control problem:

$$\min_{\mathbf{u}_i \in \mathcal{U}} \quad \int_{t_0}^{\infty} r_i(\mathbf{e}_i(\tau), \mathbf{u}_i(\tau)) d\tau \tag{5a}$$

$$\text{s.t.} \quad (1),\ (4),\ \bar{\mathbf{x}}(t_0) \in \mathcal{H}_N,\ \forall\, t \geq t_0, \tag{5b}$$

for all $i \in V_a$, where for $\mathbf{e}_i \in \mathcal{X}$, the first n_d elements denotes the goal-reaching error $\mathbf{p}_i - \mathbf{p}_i^g$ and the remaining $n - n_d$ entries are zero-padded, $\mathbf{p}_i^g \in \mathbb{P}$ is the goal position, and the immediate cost is $r_i(\mathbf{e}_i, \mathbf{u}_i) := \mathbf{e}_i^T \mathbf{Q} \mathbf{e}_i + \mathbf{u}_i^T \mathbf{R} \mathbf{u}_i$, $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$ are two symmetric, positive definite matrices.

B. Safe Optimal Controllers Synthesis via Constrained HJB

We reformulate the original problem (5) as a graph CBF-constrained HJB equation, whose solution is characterized utilizing the KKT optimality conditions with graph-dependent Lagrange multipliers.

Denote a set of admissible policies by $\mathcal{U}_a$ as in [23] where for each $\mu \in \mathcal{U}_a$, the corresponding control input additionally needs to remain within the set $\mathcal{U}$. For an admissible policy $\mathbf{u}_i \in \mathcal{U}_a$, an infinite-horizon value function is defined by $V_i(\mathbf{e}_i(t)) := \int_t^{\infty} r_i(\mathbf{e}_i(\tau), \mathbf{u}_i(\tau)) \mathrm{d}\tau$. The corresponding *Hamiltonian* is described as:

$$\begin{aligned} &H_i(\mathbf{e}_i, \mathbf{u}_i, \nabla_{\mathbf{e}_i} V_i(\mathbf{e}_i)) \\ &= \nabla_{\mathbf{e}_i} V_i(\mathbf{e}_i) \big(\mathbf{f}(\mathbf{x}_i) + \mathbf{g}(\mathbf{x}_i)\mathbf{u}_i\big) + r_i(\mathbf{e}_i, \mathbf{u}_i). \end{aligned} \tag{6}$$

Describe the optimal value function $V_i^* : \mathbb{R}^n \to \mathbb{R}$ by $V_i^*(\mathbf{e}_i(t)) = \min_{\mathbf{u}_i \in \mathcal{U}_a} \int_t^{\infty} r_i(\mathbf{e}_i(\tau), \mathbf{u}_i(\tau)) d\tau$. According to

Bellman's principle of optimality [13], the optimal value function satisfies the following HJB equation

$$\begin{aligned} 0 &= \min_{\mathbf{u}_i \in \mathcal{U}_a} H_i(\mathbf{e}_i, \mathbf{u}_i, \nabla_{\mathbf{e}_i} V_i^*(\mathbf{e}_i)) \\ &= H_i(\mathbf{e}_i, \mathbf{u}_i^*, \nabla_{\mathbf{e}_i} V_i^*(\mathbf{e}_i)), \end{aligned} \tag{7}$$

where $\mathbf{u}_i^*$ is the optimal solution of (5).

To address the problem (7), it is transformed into finding a control input $\mathbf{u}_i \in \mathcal{U}_a$ that minimizes the Hamiltonian function, subject to the graph CBF constraint in (4), for each agent i. The resulting constrained optimization problem is described by

$$\min_{\mathbf{u}_i \in \mathcal{U}_a} \quad H_i(\mathbf{e}_i, \mathbf{u}_i, \nabla_{\mathbf{e}_i} V_i^*(\mathbf{e}_i)), \tag{8a}$$

$$\text{s.t.} \quad \dot{h}(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) + \alpha(h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})) \geq 0. \tag{8b}$$

It is observed that (8) and (5) are point-wise equivalent. To address the constrained optimization problem (8) and consequently (5), we note that both the objective function (8a) and the safety constraint (8b) are convex with respect to the decision variable $\mathbf{u}_i$. Thus we can characterize the safe optimal controller $\mathbf{u}_i^*$ by applying KKT optimality conditions which are both necessary and sufficient for optimality. We have

$$\nabla_{\mathbf{u}_i^*} H_i(\mathbf{e}_i, \mathbf{u}_i^*, \nabla_{\mathbf{e}_i} V_i^*(\mathbf{e}_i)) - \lambda_i^* \nabla_{\mathbf{x}_i} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) \mathbf{g}(\mathbf{x}_i) = 0, \tag{9a}$$

$$\lambda_i^* \big(\dot{h}(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) + \alpha(h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}))\big) = 0, \tag{9b}$$

$$\dot{h}(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) + \alpha(h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})) \geq 0, \tag{9c}$$

$$\lambda_i^* \geq 0, \tag{9d}$$

where $\nabla_{\mathbf{u}_i^*} H_i(\mathbf{e}_i, \mathbf{u}_i^*, \nabla_{\mathbf{e}_i} V_i^*(\mathbf{e}_i)) := \nabla_{\mathbf{e}_i} V_i^*(\mathbf{e}_i) \mathbf{g}(\mathbf{x}_i) + 2\mathbf{u}_i^{*T} \mathbf{R}$. We derive the analytical form of safe optimal controller by solving (9a):

$$\mathbf{u}_i^* = -\frac{1}{2} \mathbf{R}^{-1} \mathbf{g}^T(\mathbf{x}_i) \big(\underbrace{\nabla_{\mathbf{e}_i}^T V_i^*(\mathbf{e}_i)}_{\text{Goal-reaching}} - \lambda_i^* \underbrace{\nabla_{\mathbf{x}_i}^T h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})}_{\text{Safety}} \big). \tag{10}$$

Substituting (10) into the complementary slackness condition (9b) yields:

$$\lambda_i^* \big(\Lambda_i + \lambda_i^* \omega_i\big) = 0, \ \forall i \in V_a, \tag{11}$$

where $\Lambda_i := -\frac{1}{2} \nabla_{\mathbf{x}_i} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) \mathbf{g}(\mathbf{x}_i) \mathbf{R}^{-1} \mathbf{g}^T(\mathbf{x}_i) \nabla_{\mathbf{e}_i}^T V_i^*(\mathbf{e}_i) + \nabla_{\mathbf{x}_i} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) \mathbf{f}(\mathbf{x}_i) + \sum_{j \in \hat{\mathcal{N}}_i^a, j \neq i} \Lambda_{ij}^e + \alpha(h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}))$, $\Lambda_{ij}^e := \nabla_{\mathbf{x}_j} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) \big(\mathbf{f}(\mathbf{x}_j) + \mathbf{g}(\mathbf{x}_j) \mathbf{u}_j\big), j \in \hat{\mathcal{N}}_i^a, j \neq i$, $\omega_i := \frac{1}{2} \nabla_{\mathbf{x}_i} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) \mathbf{g}(\mathbf{x}_i) \mathbf{R}^{-1} \mathbf{g}^T(\mathbf{x}_i) \nabla_{\mathbf{x}_i}^T h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})$. Given $\|\nabla_{\mathbf{x}_i} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i}) \mathbf{g}(\mathbf{x}_i)\| \geq \varrho$ for all $\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i} \in \mathcal{X}^M$, then $\omega_i > 0$. Under (9c) and (9d), solving (11) provides the graph-dependent Lagrange multiplier:

$$\lambda_i^* = \begin{cases} -\frac{\Lambda_i}{\omega_i}, & \text{if } \Lambda_i < 0, \\ 0, & \text{if } \Lambda_i \geq 0. \end{cases} \tag{12}$$

Adaptive trade-off mechanism: The safe optimal control (10) effectively integrates *goal-reaching control component* ($\nabla_{\mathbf{e}_i}^T V_i^*(\mathbf{e}_i)$) and *safety control component* ($\nabla_{\mathbf{x}_i}^T h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})$) via the Lagrange multiplier λ_i^*. The multiplier explicitly depends on the local interaction topology graph $\hat{\mathcal{N}}_i$ of agent i, since the set of active safety constraints is generated by its neighbor nodes. Such graph-dependent Lagrange multiplier serves as an adaptive trade-off factor that enables adaptive coordination between safety and goal-reaching control according to real-time changing interaction topology graph. Specifically, the term Λ_i characterizes the safety margin only under the goal-reaching control component. $\Lambda_i \geq 0$ indicates that the goal-reaching control naturally satisfies the CBF constraint (9c), and the multiplier λ_i^* vanishes. $\Lambda_i < 0$ indicates that the goal-reaching control threatens safety, and λ_i^* automatically activates to its optimal value $-\Lambda_i/\omega_i$. This multiply automatically increases to strengthen repulsive effects to maintain safety when neighbors are very close; and similarly, the multiply decreases accordingly to relax the strength of the safety control and prioritizes goal-reaching control when neighbor density decreases. Consequently, the proposed approach avoids the conservative trade-offs inherent in existing methods with manually tuned static parameter, heuristic gradient projection, or auxiliary variable-based trade-off designs [5, 32, 29, 33, 34], thereby reducing deadlocks and collisions in dense environments.

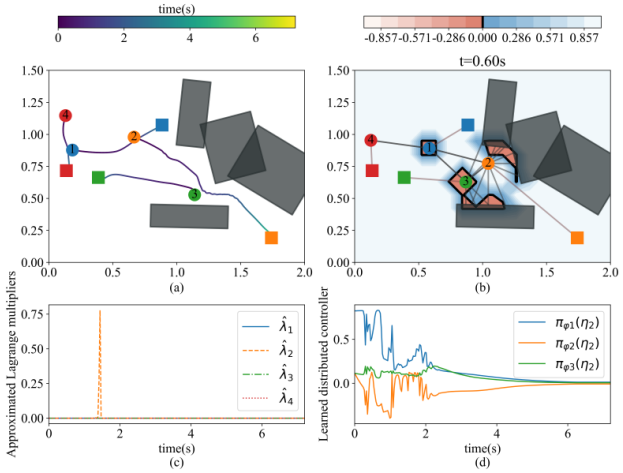

Fig. 2: Adaptive trade-off between safety and goal-reaching performance for unmanned surface vessel model of [17] in cluttered, dense environment with $2\text{m} \times 1.5\text{m}$. (a) Four agents safely navigate to their respective goals. Numbered circles denote initial positions of agents, squares with the same color as agents are their goals, and gray rectangles are static obstacles. (b) Learned graph CBF contour of agent 2 at $t = 0.6$s, where the zero-level set of learned CBF identifies the safety boundaries. (c) Approximated Lagrange multipliers $\hat{\lambda}_i, i = 1, 2, 3, 4$. (d) Learned distributed controller of agent 2. Detailed simulation setup is illustrated in Appendix-E.

As shown in Fig. 2, the evolution of $\hat{\lambda}_i$ in Fig. 2c validates the adaptive trade-off: a non-zero $\hat{\lambda}_2$ emerges when agent 2 encounters a potential collision to strength safety enforcement, while other multipliers remain at zero to preserve the unconstrained navigation optimality. It demonstrates that the adaptive trade-off mechanism enables effective safe navigation in cluttered, dense environment.

The following theorem certifies both the safety and goal-

reaching properties of the safe optimal controller (10).

Theorem 1. Consider an arbitrarily sized nonlinear MAS (1). Suppose that there exist the optimal value function $V_i^*(\mathbf{e}_i)$ of (5) being C^1 and positive definite, and a graph CBF $h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})$ satisfying $\mathcal{H}_{N,i} \subseteq \mathcal{S}_i$, $i \in V_a$ and $\|\nabla_{\mathbf{x}_i} h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})\mathbf{g}(\mathbf{x}_i)\| \geq \varrho$ with $\varrho \in \mathbb{R}_{>0}$ for all $\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i} \in \mathcal{X}^M$. Then, for any initial condition $\bar{\mathbf{x}}(t_0) \in \mathcal{H}_N$, the safe optimal controllers (10) with the graph-dependent Lagrange multipliers (12) for all $i \in V_a$ guarantee that (i) the system state remains in the safe set, i.e., $\bar{\mathbf{x}}(t) \in \mathcal{S}$ for all $t \geq t_0$, and (ii) the goal-reaching error $\mathbf{e}_i(t)$ of each agent i asymptotically converges to the zero.

Proof: See the Appendix-B. ∎

Remark 1. In Theorem 1, the assumption on $\exists V_i^* \in C^1$ is adopted mainly to simplify the presentation. We use C^1 assumption for consistency with commonly used formulations in HJB-based methods [2, 4, 13]. Theorem 1 can be easily relaxed to the case where V_i^* is locally Lipschitz, while the controller structure and the learning algorithm remain unchanged. The corresponding technical statement is provided in Remark 3 and Corollary 1 of Appendix-B. Moreover, in implementation, the desired properties of the value function and the graph CBF are promoted by the structured losses introduced in Section IV. Further details on the numerical approximation are given in Section IV and Remark 2 of Appendix-B.

To implement the analytical controller (10), we need to address the following three practical challenges: (i) the Lagrange multiplier λ_i^* used in controller (10) involves neighbor control coupling, hindering distributed execution; (ii) the value function V_i^* is difficult to be obtained; and (iii) a valid graph CBF h is unknown for unknown environments, unlike known safety constraints used in [2, 4, 1].

IV. HJB-GNN: Physics-Informed Safe Learning with GNNs

This section develops a novel HJB-GNN learning framework to approximate the safe optimal solution (10).

A. GNN-based Approximation

GNNs serve as a powerful tool for developing learning-based methods over graph-structured data in MAS. We leverage GNNs to parameterize both a graph CBF h_ϑ and a distributed safe policy π_φ with the network parameters ϑ and φ. Unlike conventional fixed-input neural network architectures [36, 20, 24, 28], GNNs inherently support variable-sized neighbor sets $\mathcal{N}_i$ and ensure generalizability to arbitrary swarm scales without requiring padding or truncation [15, 30]. We parameterize a value function $V_\theta(\mathbf{e}_i)$ using a MLP with a network parameter θ, since it only depends on goal-reaching error $\mathbf{e}_i$ not graph topology.

For h_ϑ and π_φ, we employ a graph attention-based GNN to encode local interactions within the topology graph G, augmented with a goal node for each agent and an edge connecting the agent to its goal. The input feature for agent i is denoted as $\eta_i = (\eta_{i1}, ..., \eta_{i|\mathcal{N}_i|})$, where for each $j \in \mathcal{N}_i$, the feature η_{ij} concatenates the node features to encode node type (agent, goal, or obstacle) and the edge feature containing the relative position $\mathbf{p}_j - \mathbf{p}_i$. This architecture ensures that the policy is *distributed*, as the computation of π_φ for agent i only relies on its local neighborhood $\mathcal{N}_i$. Detailed GNN architecture is illustrated in Appendix-C.

B. Training Process

In this section, we design three loss functions to train three networks $V_\theta(\mathbf{e}_i)$, $h_\vartheta(\eta_i)$, and $\pi_\varphi(\eta_i)$, respectively. We present the detailed design of each loss function below. For convenience, let $L_* = \sum_{i \in V_a} L_{*i}$ with $* \in \{v, h, u\}$.

Value Function Loss. We design the value function loss $L_{vi}(\theta, \varphi)$ consisting of *candidate Lyapunov function* loss $L_{cvi}(\theta)$ and squared *Bellman error* loss $L_{hvi}(\theta, \varphi)$, i.e.,

$$L_{vi}(\theta, \varphi) = b_{v1} L_{cvi}(\theta) + b_{v2} L_{hvi}(\theta, \varphi), \tag{13}$$

where $b_{v1},\ b_{v2} \in \mathbb{R}_{>0}$ are two constant weights. Minimizing $L_{cvi}(\theta)$ encourages $V_\theta(\mathbf{e}_i)$ to act as a candidate Lyapunov function of the system of goal-reaching error. According to the requirement that a candidate Lyapunov function be both positive definite and radially unbounded in [12], we design $L_{cvi}(\theta)$ to enforce these properties:

$$\begin{aligned} L_{cvi}(\theta) = &\ \max\big\{0, \alpha_1(\|\mathbf{e}_i\|) - V_\theta(\mathbf{e}_i)\big\} \\ &+ \max\big\{0, V_\theta(\mathbf{e}_i) - \alpha_2(\|\mathbf{e}_i\|)\big\}, \end{aligned} \tag{14}$$

where α_1, α_2 are of class $\mathcal{K}_\infty$ functions satisfying $\alpha_1(s) < \alpha_2(s)$ for any s.

Based on (6) and (7), the Bellman error $\delta_i(\theta, \varphi)$ has

$$\begin{aligned} &\delta_i(\theta, \varphi) \\ &= H_i(\mathbf{e}_i, \pi_\varphi(\eta_i), \nabla_{\mathbf{e}_i} V_\theta(\mathbf{e}_i)) - H_i(\mathbf{e}_i, \mathbf{u}_i^*, \nabla_{\mathbf{e}_i} V_i^*(\mathbf{e}_i)) \\ &= \nabla_{\mathbf{e}_i} V_\theta(\mathbf{e}_i)\big(\mathbf{f}(\mathbf{x}_i) + \mathbf{g}(\mathbf{x}_i)\pi_\varphi(\eta_i)\big) + r_i(\mathbf{e}_i, \pi_\varphi(\eta_i)). \end{aligned} \tag{15}$$

Minimizing the loss $L_{hvi}(\theta, \varphi) = \delta_i^2(\theta, \varphi)$ in (13) encourages the satisfaction of HJB optimality condition in an approximate sense, i.e., $H_i(\mathbf{e}_i, \pi_\varphi(\eta_i), \nabla_{\mathbf{e}_i} V_\theta(\mathbf{e}_i)) = 0$, guiding $V_\theta(\mathbf{e}_i)$ toward the true value function $V_i^*(\mathbf{e}_i)$. Such design also encourages the time derivative of function $V_\theta(\mathbf{e}_i)$, i.e., $\nabla_{\mathbf{e}_i} V_\theta(\mathbf{e}_i)(\mathbf{f}(\mathbf{x}_i) + \mathbf{g}(\mathbf{x}_i)\pi_\varphi(\eta_i))$, to be negative definite along the system trajectory of goal-reaching error. Together with $L_{cvi}(\theta)$, it aligns with the fact that the value function $V_i^*(\mathbf{e}_i)$ serves as a Lyapunov function for the goal-reaching error system in Theorem 1, and $V_\theta(\mathbf{e}_i)$ is trained to inherit such property. This Lyapunov function property promotes that each agent can reach to its goal position.

CBF Loss. We construct two datasets D_i^C and D_i^A consisting of input features labeled as safe and unsafe, respectively, as detailed in Appendix-D and following the procedure in [35]. The graph CBF loss $L_{hi}(\vartheta, \varphi)$ encourages the satisfaction of safety constraint (4) and the safety requirement from

$\mathcal{H}_{N,i} \subseteq \mathcal{S}_i$ in Theorem 1, i.e.,

$$\begin{aligned} & L_{hi}(\vartheta, \varphi) \\ = & \sum_{\eta_i \in D_i^C} \max\left\{0, \varsigma - h_\vartheta(\eta_i)\right\} + \sum_{\eta_i \in D_i^A} \max\left\{0, \varsigma + h_\vartheta(\eta_i)\right\} \\ & + b_h \sum_{\eta_i} \max\left\{0, \varsigma - \dot{h}_\vartheta(\eta_i) - \alpha(h_\vartheta(\eta_i))\right\}, \end{aligned} \tag{16}$$

where $\varsigma \in \mathbb{R}_{>0}$ is a constant weight to encourage these conditions to strictly be satisfied, and $b_h \in \mathbb{R}_{>0}$ is a constant weight. Minimizing the loss (16) encourages that each agent maintains safety distances with other agents and obstacles.

Controller Loss. We design the controller loss function $L_{ui}(\theta, \vartheta, \varphi)$ as:

$$L_{ui}(\theta, \vartheta, \varphi) = L_{\pi i}(\theta, \vartheta, \varphi) + L_{hi}(\vartheta, \varphi) + L_{hvi}(\theta, \varphi). \tag{17}$$

Here, minimizing the CBF loss $L_{hi}(\vartheta, \varphi)$ promotes the controller $\pi_\varphi(\eta_i)$ to satisfy the safety constraints. Minimizing $L_{hvi}(\theta, \varphi)$ enforces the satisfaction of HJB optimality condition and guides the time derivative of value function to be negative definite and the controller to be stabilizing.

We introduce the loss $L_{\pi i}(\theta, \vartheta, \varphi)$ to minimize the deviation between $\pi_\varphi(\eta_i)$ and the approximated controller $\hat{\mathbf{u}}_i$ of safe optimal controller (10), without requiring the QP-CBF controller in [35]:

$$L_{\pi i}(\theta, \vartheta, \varphi) = b_\pi \|\pi_\varphi(\eta_i) - \hat{\mathbf{u}}_i\|, \tag{18}$$

where $b_\pi \in \mathbb{R}_{>0}$ a constant weight, the approximated controller $\hat{\mathbf{u}}_i$ of (10) is

$$\hat{\mathbf{u}}_i = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{g}^T(\mathbf{x}_i)\big(\underbrace{\nabla_{\mathbf{e}_i} V_\theta(\mathbf{e}_i)}_{\text{Goal-reaching}} - \hat{\lambda}_i \underbrace{\nabla_{\eta_i} h_\vartheta(\eta_i)}_{\text{Safety}}\big), \tag{19}$$

and the approximated Lagrange multiplier $\hat{\lambda}_i$ is defined in (11) when replacing $V_i^*(\mathbf{e}_i)$ and $h(\bar{\mathbf{x}}_{\hat{\mathcal{N}}_i})$ with $V_\theta(\mathbf{e}_i)$ and $h_\vartheta(\eta_i)$, respectively. By minimizing the deviation between $\pi_\varphi(\eta_i)$ and $\hat{\mathbf{u}}_i$, the loss (18) guides $\pi_\varphi(\eta_i)$ to inherit the safety, goal-reaching performance, and optimality structure encoded in the KKT-based controller. This design achieves the adaptive trade-off between safety and goal-reaching performance via $\hat{\lambda}_i$, and distributed control $\pi_\varphi(\eta_i)$ instead of centralized control (10).

HJB-GNN Algorithm. The overall algorithm is given in Algorithm 1. We perform the training by minimizing the corresponding loss functions via gradient descent, during which θ, ϑ, and φ are updated correspondingly. Specifically, we can randomly initialize θ, ϑ and conduct a warm-up phase by iteratively minimizing the value function loss (13) to yield an admissible initial controller π_φ. Then, we adopt two phases in each iteration: **(i) Phase 1:** With the controller $\pi_\varphi(\eta_i)$ fixed, the value function and graph CBF are updated to evaluate the current controller by minimizing the loss $L_v(\theta, \varphi)$ and $L_h(\theta, \varphi)$, respectively. Phase 1 terminates when the empirical expectation of the squared Bellman error $\mathbb{E}_{\mathbf{e}_i \sim D_v} \delta_i^2(\theta, \varphi)$ for all $i \in V_a$ is reduced. **(ii) Phase 2:** With the value function fixed, the controller is improved by minimizing the controller loss, while the graph CBF is further refined. This phase encourages the controller to be stabilizing and safe with respect to the current value function. Phase 2 terminates when the empirical expectation of the Hamiltonian becomes non-positive, i.e., $\mathbb{E}_{\mathbf{e}_i \sim D_v} \dot{V}_\theta(\mathbf{e}_i(t)) < 0$ for all $i \in V_a$, indicating the controller to be stabilizing. By alternating between these two phases, the algorithm progressively yields a distributed safe policy $\pi_\varphi(\eta_i)$ for multi-agent navigation.

Algorithm 1 HJB-GNN Learning Algorithm for Multi-Agent Navigation.

Initialization: Network parameters θ, ϑ, φ, the compact set $\mathcal{X}$, and the number of samples n_s
for each iteration **do**
  Collect D_i^A and D_i^C using policy $\pi_\varphi(\eta_i)$, $\forall i \in V_a$
  Collect D_v via uniformly sampling n_s data from $\mathcal{X}$
  # Phase 1: update $V_\theta(\mathbf{e}_i)$ and $h_\vartheta(\eta_i)$ under fixed $\pi_\varphi(\eta_i)$
  repeat
    Update θ by minimizing $L_v(\theta, \varphi)$
    Update ϑ by minimizing $L_h(\vartheta, \varphi)$
  until $\mathbb{E}_{\mathbf{e}_i \sim D_v} \delta_i^2(\theta, \varphi)$ is reduced for all $i \in V_a$
  # Phase 2: update $\pi_\varphi(\eta_i)$ and $h_\vartheta(\eta_i)$ under fixed $V_\theta(\mathbf{e}_i)$
  repeat
    Update φ by minimizing $L_u(\theta, \vartheta, \varphi)$
    Update ϑ by minimizing $L_h(\vartheta, \varphi)$
  until $\mathbb{E}_{\mathbf{e}_i \sim D_v} \dot{V}_\theta(\mathbf{e}_i(t)) < 0$ for all $i \in V_a$
end for

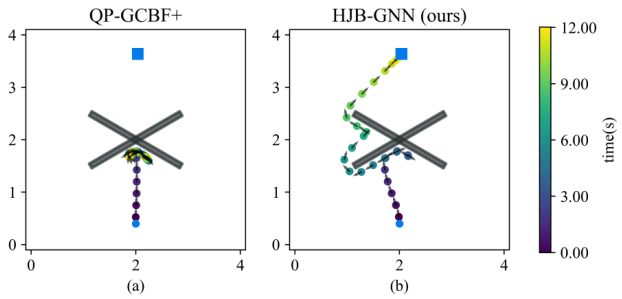

Fig. 3: Comparison between QP-GCBF+ of [35] in (a) and our HJB-GNN in (b). An agent (a gradient-colored circle) aims to reach the goal (blue square) while avoiding static obstacles (gray rectangles) within a $4\text{m} \times 4\text{m}$ region. Black arrow indicates the velocity direction of agent. The QP-GCBF+ has deadlock near obstacles, and the HJB-GNN achieves goal-reaching without collision.

Reduced Deadlocks. Our HJB-GNN algorithm can reduce deadlocks that can be caused by the use of short-horizon QP solvers [11, 16, 25, 35]. We present a comparison between state-of-the-art QP-GCBF+ method in [35] and our HJB-GNN approach in Fig. 3. The QP-GCBF+ method in [35] suffers from deadlocks near obstacles, our HJB-GNN controller ensures a collision-free trajectory to the goal.

V. Simulations and Hardware Experiments

In this section, we design extensive simulations and hardware experiments to answer the three main questions:

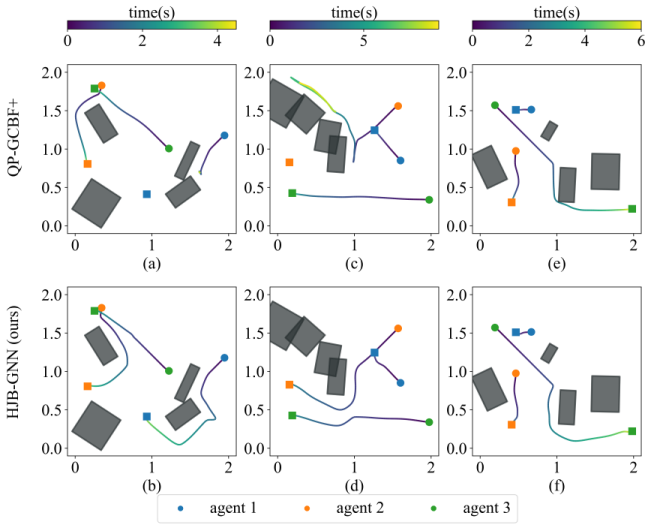

Fig. 4: Comparisons between the QP-GCBF+ method ((a), (c), and (e)) in [35] and our HJB-GNN approach ((b), (d), and (f)) under three cases with four agents (circles indicating their initial positions) and four static obstacles (gray rectangles) in a $2\text{m} \times 2\text{m}$ region. Agent 1 in (a) and agent 2 in (c) experience deadlock, and agent 3 in (e) collides with an obstacle. All agents safely reach their respective goals (squares with the same colors as agents) in (b), (d), and (f).

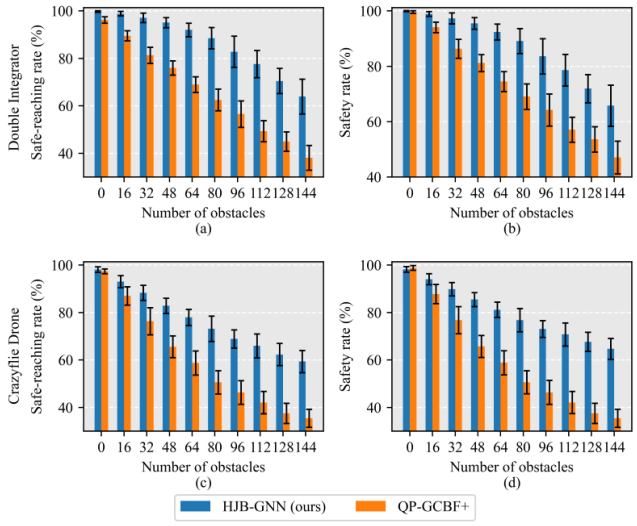

Fig. 5: Safe-reaching rates and safety rates for increasing numbers of obstacles and 256 agents under fixed area width. (a)-(b): 2D double integrator, and (c)-(d): 3D Crazyflie drone.

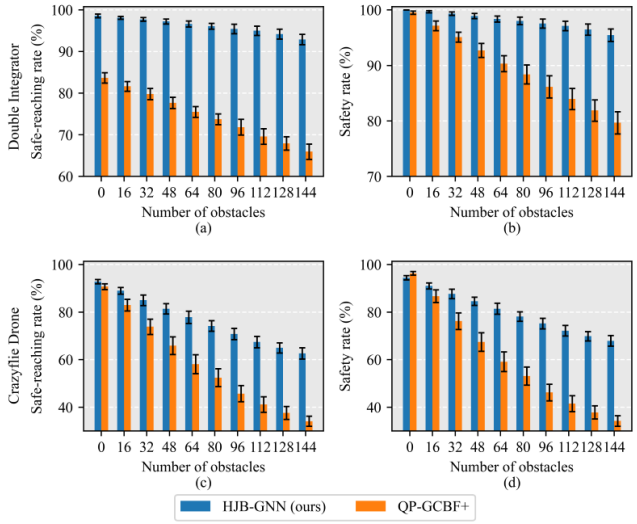

Fig. 6: Safe-reaching rates and safety rates for increasing numbers of obstacles and 1024 agents under fixed area width. (a)-(b): 2D double integrator, and (c)-(d): 3D Crazyflie drone.

(Q1) Does the learned policy from HJB-GNN algorithm reduce deadlocks and collisions in existing short-horizon QP-CBF methods with predefined nominal controllers?

(Q2) How well does the learned policy from HJB-GNN algorithm scale to unseen dense environments and generalize to much larger MAS?

(Q3) Can HJB-GNN algorithm is robust enough for effective deployment on physical robotic platforms?

Details on the simulations and hardware implementation, hyperparameters, and more evaluation results are given in Appendixes-E and -F.

A. Simulations Experiments

We compare our HJB-GNN approach and state-of-the-art QP-GCBF+ method in [35] on both 2D linear double integrator and 3D nonlinear Crazyflie drone systems as described in [35]. Two metrics are used to evaluate performance: (i) **safety rate**, defined as the proportion of agents that avoid all collisions throughout the simulation; and (ii) **safe-reaching rate**, defined as the proportion of agents that both remain collision-free and successfully reach their goals. For each simulation, the performance is assessed by randomly choosing 32 initial positions of agents and goal positions, and safety rates and safe-reaching rates are reported as the mean and standard deviation over these 32 instances. All experiments are trained with 8 agents and 8 static obstacles.

(Q1): HJB-GNN approach reduces deadlocks and collisions in the baseline QP-GCBF+ [35]. We conduct three comparative experiments in dense, cluttered environments, shown in Fig. 4. The QP-GCBF+ method suffers from deadlocks (see Agent 1 in (a) and Agent 2 in (c) of Fig. 4) or collisions (Agent 3 in (e) of Fig. 4). In contrast, our HJB-GNN approach ensures that all agents safely reach their goals, shown in Fig. 4 (b), (d), and (f). In the baseline, the failures stem from the structural limitations caused by short-horizon optimization and non-safety-aware fixed nominal controllers, which restrict the coordination between safety and goal-reaching, thereby easily causing deadlocks and collisions in dense, cluttered environments. In the HJB-GNN, we leverage an infinite-horizon optimal formulation and the adaptive trade-off mechanism, which ensure that the learned policy provides more forward-looking and flexible decision-making, thereby

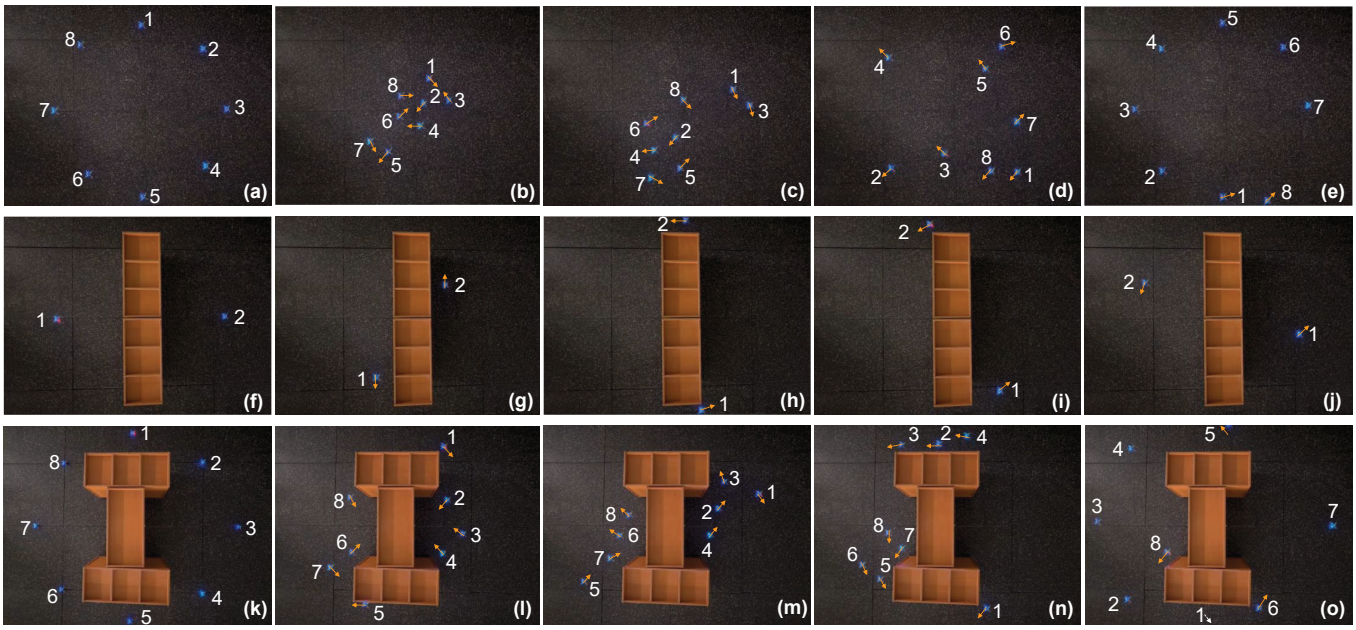

Fig. 7: **Hardware experiments on three antipodal position-swapping tasks.** From left to right: key frames of trajectories across three experiments. Orange arrows denote velocity directions. (a)-(e): Eight Crazyflie drones in an obstacle-free environment. (f)-(j): Two drones navigating around a long static obstacle. (k)-(o): Eight drones navigating among large non-convex obstacles. white dashed arrow in (o) indicates that Drone 1 has reached its goal.

reducing deadlocks and collisions.

(Q2): The learned policy can scale to unseen dense environments and generalize to much larger MAS. We conduct two sets of simulations by fixing number of agents while increasing obstacle density, as illustrated in Fig. 5 (256 agents) and Fig. 6 (1024 agents). The results demonstrate that HJB-GNN consistently outperforms the QP-GCBF+ baseline in both safety and safe-reaching rates. Notably, the performance advantage becomes more pronounced as obstacle density intensifies, primarily due to the rapid degradation of the baseline method. This failure in the baseline stems from its reliance on precomputed nominal controllers, which restrict the flexibility to adaptively reconcile two competing objectives. Furthermore, its short-horizon optimization often induces deadlocks and overly-conservative behaviors in high-density scenarios. In contrast, our approach effectively overcomes these limitations by integrating infinite-horizon optimal navigation with a principled adaptive trade-off mechanism. The HJB-GNN approach significantly enhances the policy's scalability and its ability to generalize to large-scale MAS in previously unseen, high-density environments.

B. Hardware Experiments

We demonstrate the applicability and robustness of HJB-GNN by four sets of challenging antipodal position-swapping tasks using a swarm of Crazyflie 2.1 platforms [3], shown in Figs. 1 and 7.

The control architecture for the Crazyflie swarm used in this paper follows an offboard-onboard split, considering the limited computation resources of Crazyflies. The distributed safe policy is running offboard using a PC that is connected to the Crazyflies using Crazyradio PA. [4]. We use the Crazyswarm package [5] to communicate with Crazyflies based on ROS1, which allows for sending full state control commands to the Crazyflies. Moreover, the learned policy requires only $\sim$ 1.5 MB memory per agent and a single forward pass at inference, with no online optimization. The computational cost is minimal; for typical problem sizes, inference can be executed in a few milliseconds on embedded platforms such as NVIDIA Jetson Orin Nano.

(Q3): HJB-GNN algorithm can be effectively deployed on physical Crazyflie drone platforms. All drones are uniformly initialized on a circle of radius $1\mathrm{m}$ in (a) of Fig. 1 and (a), (f) of Fig. 7, and $1.4\mathrm{m}$ in (k) of Fig. 7. *(i) Dense agent environment:* Fig. 7 (a) contains typical inter-agent interactions and serves as a representative way to evaluate both safety and goal-reaching capabilities. All drones successfully reach their respective antipodal goal positions without collisions throughout the process, see Fig. 7 (b)-(e). *(ii) Static large, non-convex obstacles:* Fig. 7 (f) evaluates the HJB-GNN's capability to avoid deadlocks and ensure safety in the presence of long obstacle. Fig. 7 (k) increases the difficulty due to the risk of getting trapped or experiencing deadlocks near concave corners. From Fig. 7 (g)-(j) and Fig. 7 (l)-(o), all drones safely reach their respective goal positions without collisions and deadlocks, highlighting the robustness of our HJB-GNN approach in cluttered environments with complex

[3]https://www.bitcraze.io/products/old-products/crazyflie-2-1/

[4]https://www.bitcraze.io/products/crazyradio-pa/

[5]https://crazyswarm.readthedocs.io/en/latest/

static obstacle geometries. *(iii) Large dynamic obstacle:* The experiment in Fig. 1 (a) contains both a static obstacle and a large dynamic obstacle. Our control policy is trained in environments with only static obstacles, since we consider that obstacle velocities are unavailable due to limited sensing capability and thus didn't utilize them. To evaluate the scalability of our HJB-GNN approach, we introduce a dynamic obstacle whose motion is unknown to the agents and whose size is significantly larger than that of a drone. By Fig. 1 (b)-(e), all drones successfully avoid obstacles and reach their goal positions, indicating the scalability of HJB-GNN approach to previously unseen dynamic obstacle environments.

VI. Conclusions

This paper has proposed HJB-GNN, a novel constrained infinite-horizon optimization-guided learning framework for safe multi-agent navigation in unknown, cluttered, and dense environments. By utilizing the KKT optimality conditions, the proposed framework has solved constrained HJB optimal control problem with graph CBF constraints, and derived graph-dependent Lagrange multipliers to achieve adaptive trade-off between safety and goal-reaching behaviors, depending on changing interaction topology graph. The resulting HJB-GNN algorithm has jointly learned a value function, a graph CBF, and a distributed policy in an end-to-end manner, without relying on short-horizon QP solvers, predefined nominal controllers, or heuristic safety-task trade-off designs. Such algorithm provided a centralized training and distributed execution manner. Extensive simulation results demonstrated that HJB-GNN approach outperforms the state-of-the-art baseline in terms of safety, scalability, and generalizability, particularly in densely cluttered environments. Moreover, real-world experiments with Crazyflie drone swarms on challenging antipodal position-swapping tasks validated the practical applicability of the HJB-GNN approach.

Acknowledgment

This work was supported by the Singapore Ministry of Education Tier 2 Academic Research Fund (T2EP20123-0037, T2EP20224-0035).

References

[1] H. Almubarak, E. A. Theodorou, and N. Sadegh. HJB based optimal safe control using control barrier functions. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6829–6834. IEEE, 2021.

[2] S. Bandyopadhyay and S. Bhasin. Lagrangian-based online safe reinforcement learning for state-constrained systems. *Automatica*, 179:112458, 2025.

[3] J. J. Choi, J. J. Aloor, J. Li, M. G. Mendoza, H. Balakrishnan, and C. J. Tomlin. Resolving conflicting constraints in multi-agent reinforcement learning with layered safety. In *Proceedings of Robotics: Science and Systems*, 2025.

[4] M. H. Cohen and C. Belta. Safe exploration in model-based reinforcement learning using control barrier functions. *Automatica*, 147:110684, 2023.

[5] M. Dawood, S. Pan, N. Dengler, S. Zhou, A. P. Schoellig, and M. Bennewitz. Safe multi-agent reinforcement learning for behavior-based cooperative navigation. *IEEE Robotics and Automation Letters*, 10(6):6256–6263, 2025.

[6] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan. Learning safe, generalizable perception-based hybrid control with certificates. *IEEE Robotics and Automation Letters*, 7(2):1904–1911, 2022.

[7] L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. CRC Press, 2025. doi: 10.1201/9781003583004.

[8] H. Farivarnejad, A. S. Lafmejani, and S. Berman. Local navigation-like functions for safe robot navigation in bounded domains with unknown convex obstacles. *Automatica*, 161:111452, 2024.

[9] Z. Gao, G. Yang, J. Bayrooti, and A. Prorok. Provably safe online multi-agent navigation in unknown environments. In *8th Annual Conference on Robot Learning*, 2024.

[10] F. Huang, X. Chen, and Z. Chen. Cooperative target enclosing control for multiple unmanned surface vehicles with unknown dynamics and safety assurance. *IEEE Transactions on Intelligent Vehicles*, 10(3):1678–1692, 2025.

[11] M. Jankovic, M. Santillo, and Y. Wang. Multiagent systems with CBF-based controllers: Collision avoidance and liveness from instability. *IEEE Transactions on Control Systems Technology*, 32(2):705–712, 2024.

[12] H. K. Khalil. *Nonlinear systems*. Prentice-Hall, New Jersey, 1996.

[13] F. L. Lewis, D. Vrabie, and V. L. Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[14] A. Lin, S. Peng, and S. Bansal. One filter to deploy them all: Robust safety for quadrupedal navigation in unknown environments. *IEEE Transactions on Robotics*, 2025. doi: 10.1109/TRO.2025.3644957.

[15] Y. Ma, Q. Khan, and D. Cremers. Multi agent navigation in unconstrained environments using a centralized attention based graphical neural network controller. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2893–2900, 2023.

[16] P. Mestres, C. Nieto-Granda, and J. Cortés. Distributed safe navigation of multi-agent systems using control barrier function-based controllers. *IEEE Robotics and Automation Letters*, 9(7):6760–6767, 2024.

[17] R. Namerikawa, A. Wiltz, F. Mehdifar, T. Namerikawa, and D. V. Dimarogonas. On the equivalence between prescribed performance control and control barrier functions. In *2024 American Control Conference (ACC)*, pages 2458–2463, 2024.

[18] H. Parwana, A. Mustafa, and D. Panagou. Trust-based rate-tunable control barrier functions for non-cooperative multi-agent systems. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2222–2229. IEEE, 2022.

[19] S. Safaoui, A. P. Vinod, A. Chakrabarty, R. Quirynen, N. Yoshikawa, and S. Di Cairano. Safe multiagent motion planning under uncertainty for drones using filtered reinforcement learning. *IEEE Transactions on Robotics*, 40:2529–2542, 2024.

[20] A. D. Saravanos, Y. Aoyama, H. Zhu, and E. A. Theodorou. Distributed differential dynamic programming architectures for large-scale multiagent control. *IEEE Transactions on Robotics*, 39(6):4387–4407, 2023.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[22] X. Tan, C. Liu, K. H. Johansson, and D. V. Dimarogonas. A continuous-time violation-free multiagent optimization algorithm and its applications to safe distributed control. *IEEE Transactions on Automatic Control*, 70(8):5114–5128, 2025.

[23] K. G. Vamvoudakis and F. L. Lewis. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

[24] A. P. Vinod, S. Safaoui, T. H. Summers, N. Yoshikawa, and S. Di Cairano. Decentralized, safe, multiagent motion planning for drones under uncertainty via filtered reinforcement learning. *IEEE Transactions on Control Systems Technology*, 32(6):2492–2499, 2024.

[25] L. Wang, A. D. Ames, and M. Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[26] Z. Wang, T. Hu, and L. Long. Multi-UAV safe collaborative transportation based on adaptive control barrier function. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(11):6975–6983, 2023.

[27] S. Wu and L. Long. Obstacle avoidance and safe coverage of moving domains for multiagent systems via adaptive control barrier function. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 55(7):5080–5090, 2025.

[28] B. Yan, P. Shi, C. P. Lim, Y. Sun, and R. K. Agarwal. Security and safety-critical learning-based collaborative control for multiagent systems. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2777–2788, 2025.

[29] Y. Yang, Y. Zhang, W. Zou, J. Chen, Y. Yin, and S. E. Li. Synthesizing control barrier functions with feasible region iteration for safe reinforcement learning. *IEEE Transactions on Automatic Control*, 69(4):2713–2720, 2024.

[30] C. Yu, H. Yu, and S. Gao. Learning control admissibility models with graph neural networks for multi-agent navigation. In *Conference on Robot Learning*, pages 934–945. PMLR, 2023.

[31] L. Zhang, R. Zhang, T. Wu, R. Weng, M. Han, and Y. Zhao. Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5435–5444, 2021.

[32] S. Zhang, K. Garg, and C. Fan. Neural graph control barrier functions guided distributed collision-avoidance multi-agent control. In *Conference on Robot Learning*, pages 2373–2392. PMLR, 2023.

[33] S. Zhang, O. So, M. Black, and C. Fan. Discrete GCBF proximal policy optimization for multi-agent safe optimal control. In *Conference on Learning Representations*, 2025.

[34] S. Zhang, O. So, M. Black, Z. Serlin, and C. Fan. Solving multi-agent safe optimal control with distributed epigraph form MARL. In *Proceedings of Robotics: Science and Systems*, 2025.

[35] S. Zhang, O. So, K. Garg, and C. Fan. GCBF+: A neural graph control barrier function framework for distributed safe multiagent control. *IEEE Transactions on Robotics*, 41:1533–1552, 2025.

[36] X. Zhang, W. Pan, C. Li, X. Xu, X. Wang, R. Zhang, and D. Hu. Toward scalable multirobot control: Fast policy learning in distributed MPC. *IEEE Transactions on Robotics*, 41:1491–1512, 2025.